Short replies to the questions mentioned in the task.

**How you went about building this model?**

First, the data was explored to identify imperfections and determine the information that could be used to build the model. Checks for duplicates, consistency in data types, handling of missing values, and feature engineering techniques were applied to investigate dependencies in the data and select features for training.

The task was defined as a multiclass classification problem with limited labeled data and imbalanced classes. To address these challenges, a semi-supervised training approach using XGBClassifier was chosen. The labeled data was split into training, validation, and test sets. During training, class-specific F1 scores were tracked to monitor performance on imbalanced classes and to account for noise in the data. Improving the model’s performance on minority classes was prioritized over maximizing overall accuracy.

Finally, the trained model was evaluated on the test set by calculating precision, recall, and F1 score per class, along with averages and overall accuracy. Based on these results, conclusions were drawn and further steps for model improvement were outlined.

Raw Data

    ↓

Data Curation and Feature Engineering

    ↓

Labeled/Unlabeled Data Split

    ↓

Train/Validation/Test Split (70/10/20)

    ↓

Self-Training Loop (up to 20 iterations):

    ├── XGBoost Training with Class Weights

    ├── Validation Performance Monitoring

    ├── Pseudo-Label Generation (Confidence Thresholds)

    ├── High-Confidence Sample Selection

    └── Training Set Augmentation

    ↓

Final Model Evaluation on Test Set

**Why you used this type of model?**

The modeling strategy was designed to address two main issues in the data: limited labeled data (10% of the dataset) and imbalanced class distribution (Chocolate/Vanilla/Strawberry = 6475/1922/1589).

A semi-supervised training approach was applied, where the training set was gradually expanded by adding samples that the model predicted with a predefined level of confidence. Class-specific confidence thresholds were used to reduce bias when adding new labels.

XGBClassifier was chosen for training because its sequential trees focus on misclassified examples (e.g., minority classes), it does not require numerical feature scaling, supports categorical features natively in XGBoost (≥1.5), and allows class weighting to further address imbalance.

**Summarise the interesting findings of your results?**

Data analysis:

* The labelled dataset is limited and skewed towards the "Chocolate" class.
* Inconsistent categorical entries were identified (e.g., MOBILE vs Mobile, MobileAPP vs mobile, etc.).
* Over 1% of records show the first bet occurring before registration, indicating data quality issues.
* Three distinct peaks in registrations were observed, likely corresponding to major sporting events.

Modelling results:

With an overall accuracy of 0.52, the model appears to be overfitting to the majority class (Chocolate) and struggling with class imbalance. Performance is weakest for the Strawberry class, which shows the lowest precision and recall. The semi-supervised approach slightly improved F1 scores across all classes, with the most noticeable improvement in the Chocolate class. Overall accuracy increased to 0.60. Further model tuning should be performed.

1 iteration

| Class | Precision | Recall | F1-score | Support |
| --- | --- | --- | --- | --- |
| Chocolate | 0.76 | 0.50 | 0.60 | 1282 |
| Strawberry | 0.29 | 0.42 | 0.34 | 313 |
| Vanilla | 0.36 | 0.65 | 0.47 | 381 |
| Accuracy |  |  | 0.52 | 1976 |
| Macro avg | 0.47 | 0.52 | 0.47 | 1976 |
| Weighted avg | 0.61 | 0.52 | 0.54 | 1976 |
|  |  |  |  |  |

**A diagram of a test set

AI-generated content may be incorrect.**

Semi-supervised (7 iterations)

| Class | Precision | Recall | F1-score | Support |
| --- | --- | --- | --- | --- |
| Chocolate | 0.73 | 0.69 | 0.71 | 1282 |
| Strawberry | 0.34 | 0.35 | 0.35 | 313 |
| Vanilla | 0.45 | 0.53 | 0.49 | 381 |
| Accuracy |  |  | 0.60 | 1976 |
| Macro avg | 0.51 | 0.52 | 0.51 | 1976 |
| Weighted avg | 0.62 | 0.60 | 0.61 | 1976 |

**A diagram of a test set

AI-generated content may be incorrect.**

**What else would you do if you had more time?**

* Creating new categorical features with info about big sport events and age groups.
* Better handling of inconsistent categorical entries and negative DaysToFirstBet (clarification info is needed).
* Limiting the number of pseudo-labelled data added to the training set per class. Dynamic confidence threshold for the classes. More aggressive weighting.
* Using oversampling for minority classes.
* Hyperparameter tuning.